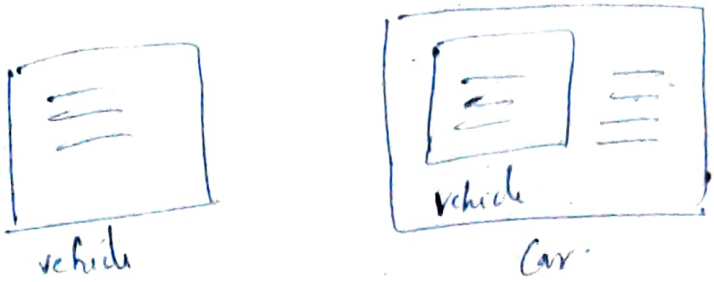
4) Inheritance: Order of Constructor Destructor




```
#include <iostream>
using namespace std;
#include "vehicle.cpp"
#include "car.cpp"
int main() {
    Vehicle v;
    v.color = "Blue";
    // ~~v.increase()~~
    Car c;
    c.color = "Black";
    c.numGear = 5;
```

→ Firstly, vehicle class default constructor will be called. Completely then car class constructor is involved.

A Base class

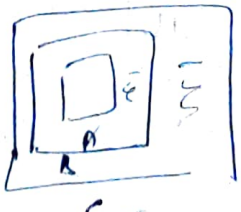
B child class

C child class

A a; → only A class constructor will be invoked

B b; → First A class constructor is invoked followed B class constructor

C c;



A() → First
B() → Second
C() → third
} implicitly

* In case of destructor invocation is just opposite to the constructor invocation

Vehicle

↓

Car

Implicitly the code of Car constructor is

```
Car() : Vehicle() {
        cout << "Car default constructor is called!" << endl;
}
```

→ This is called implicitly when there is no parameterized constructor is defined in vehicle class

→ Else we need to write above Car constructor syntax explicitly

Suppose if vehicle has a parameterized constructor

```
Car() : Vehicle(5) {
        cout << "Car constructor called" << endl;
}
```

Vehicle parameterized constructor is called